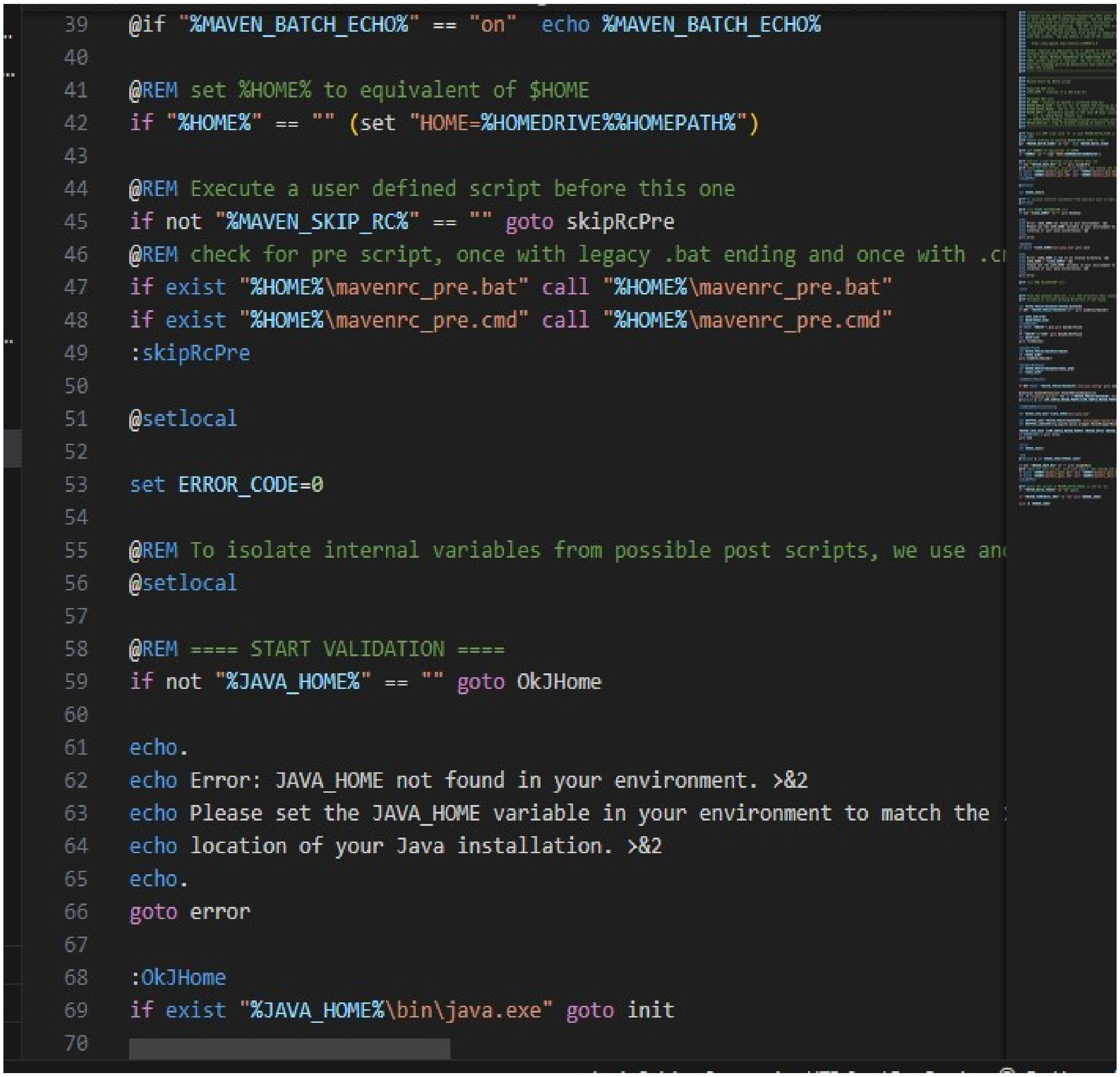
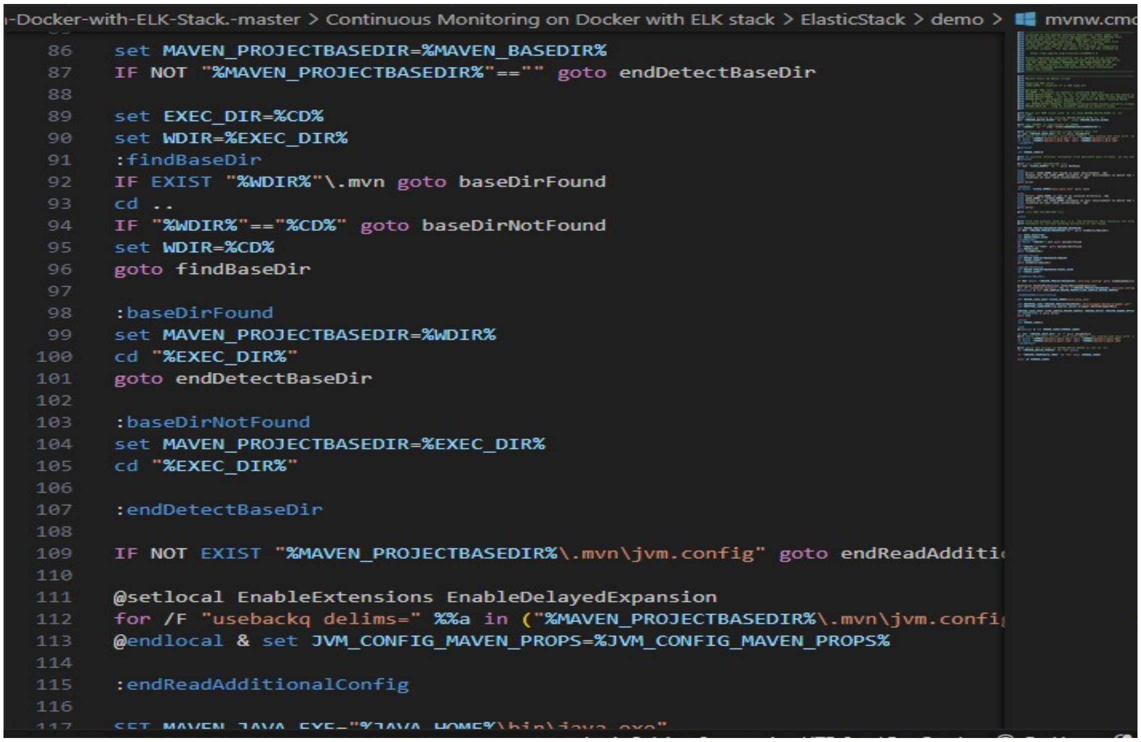
ScreenShot of the Source Code

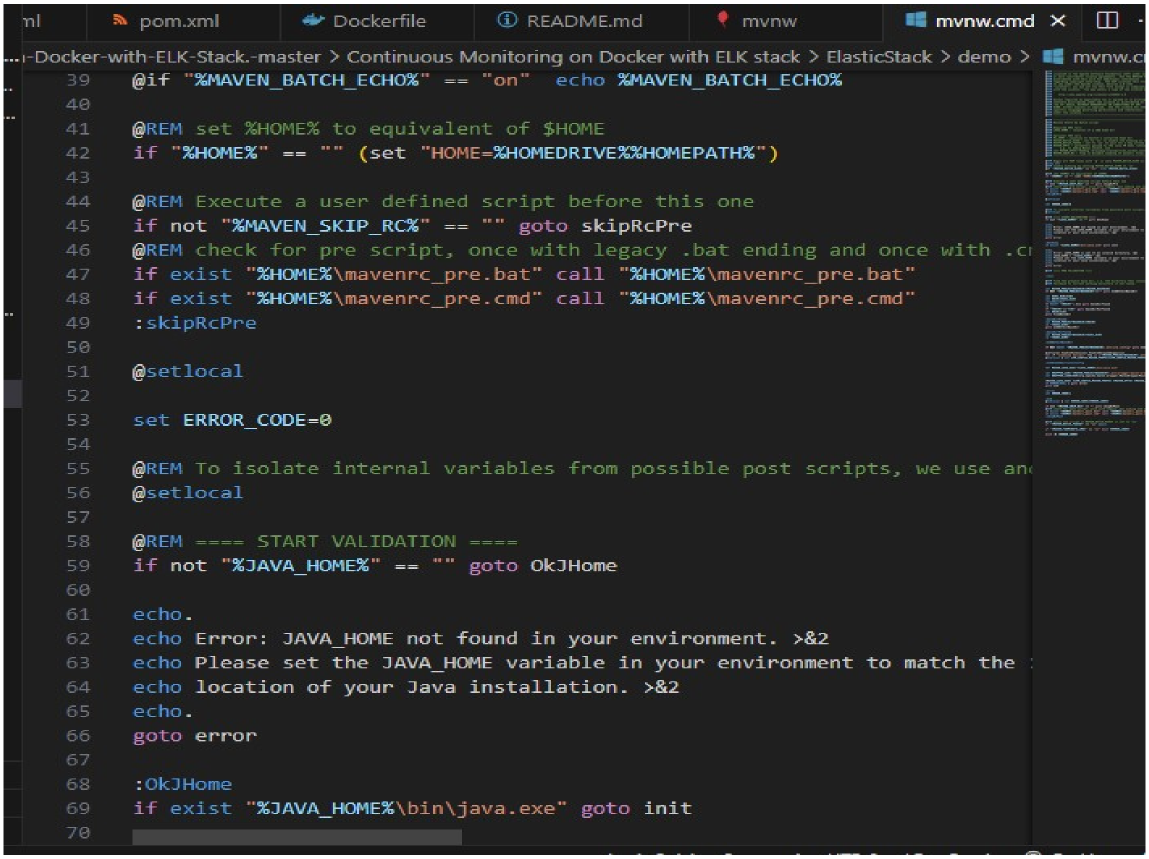
Mvnw.cmd



```
@if "%MAVEN_BATCH_ECHO%" == "on"  echo %MAVEN_BATCH_ECHO%

@REM set %HOME% to equivalent of $HOME
if "%HOME%" == "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")

@REM Execute a user defined script before this one
if not "%MAVEN_SKIP_RC%" == "" goto skipRcPre
@REM check for pre script, once with legacy .bat ending and once with .c
if exist "%HOME%\mavenrc_pre.bat" call "%HOME%\mavenrc_pre.bat"
if exist "%HOME%\mavenrc_pre.cmd" call "%HOME%\mavenrc_pre.cmd"
:skipRcPre

@setlocal

set ERROR_CODE=0

@REM To isolate internal variables from possible post scripts, we use an
@setlocal

@REM ==== START VALIDATION ====
if not "%JAVA_HOME%" == "" goto OkJHome

echo.
echo Error: JAVA_HOME not found in your environment. >&2
echo Please set the JAVA_HOME variable in your environment to match the
echo location of your Java installation. >&2
echo.
goto error

:OkJHome
if exist "%JAVA_HOME%\bin\java.exe" goto init

```

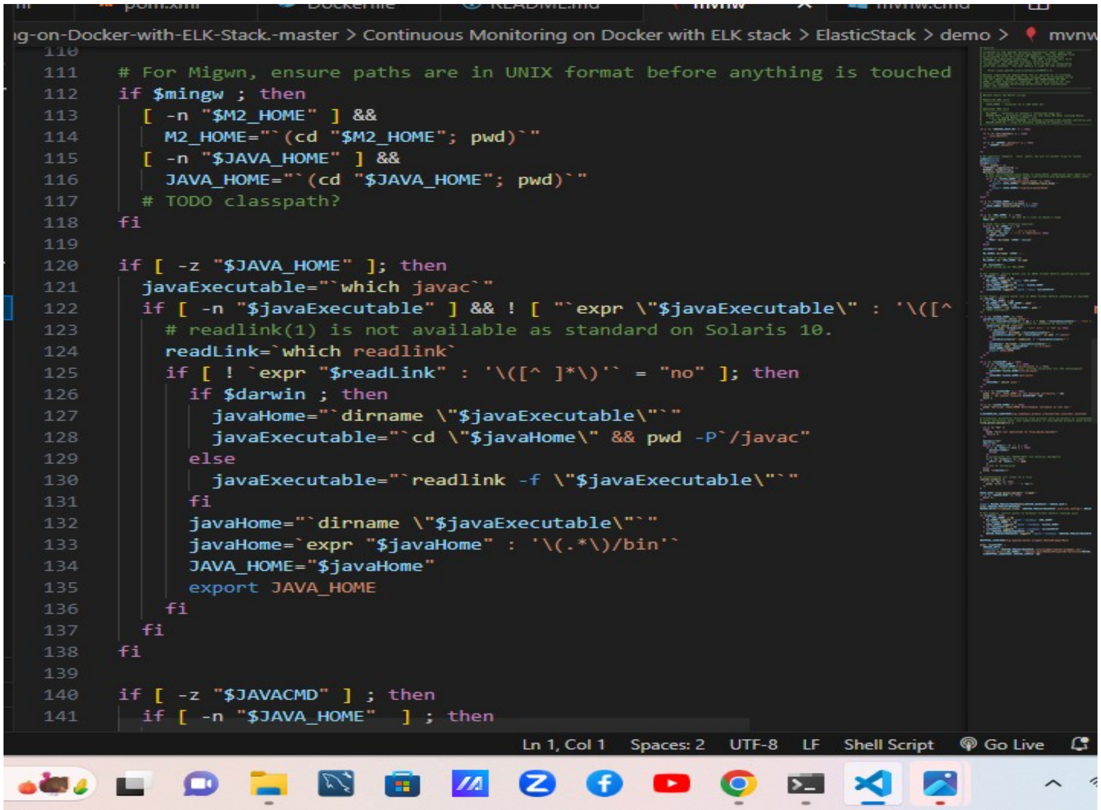

-Docker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > mvnw.cmc

```
set MAVEN_PROJECTBASEDIR=%MAVEN_BASEDIR%
IF NOT "%MAVEN_PROJECTBASEDIR%"=="" goto endDetectBaseDir

set EXEC_DIR=%CD%
set WDIR=%EXEC_DIR%
:findBaseDir
IF EXIST "%WDIR%"\.mvn goto baseDirFound
cd ..
IF "%WDIR%"=="%CD%" goto baseDirNotFound
set WDIR=%CD%
goto findBaseDir

:baseDirFound
set MAVEN_PROJECTBASEDIR=%WDIR%
cd "%EXEC_DIR%"
goto endDetectBaseDir

:baseDirNotFound
set MAVEN_PROJECTBASEDIR=%EXEC_DIR%
cd "%EXEC_DIR%"

:endDetectBaseDir

IF NOT EXIST "%MAVEN_PROJECTBASEDIR%\.mvn\jvm.config" goto endReadAdditi

@setlocal EnableExtensions EnableDelayedExpansion
for /F "usebackq delims=" %%a in ("%MAVEN_PROJECTBASEDIR%\.mvn\jvm.confi
@endlocal & set JVM_CONFIG_MAVEN_PROPS=%JVM_CONFIG_MAVEN_PROPS%

:endReadAdditionalConfig

SET MAVEN_JAVA_EXE="%JAVA_HOME%\bin\java.exe"
```


ml
pom.xml
Dockerfile
README.md
mvnw
mvnw.cmd ×
...-Docker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > mvnw.c

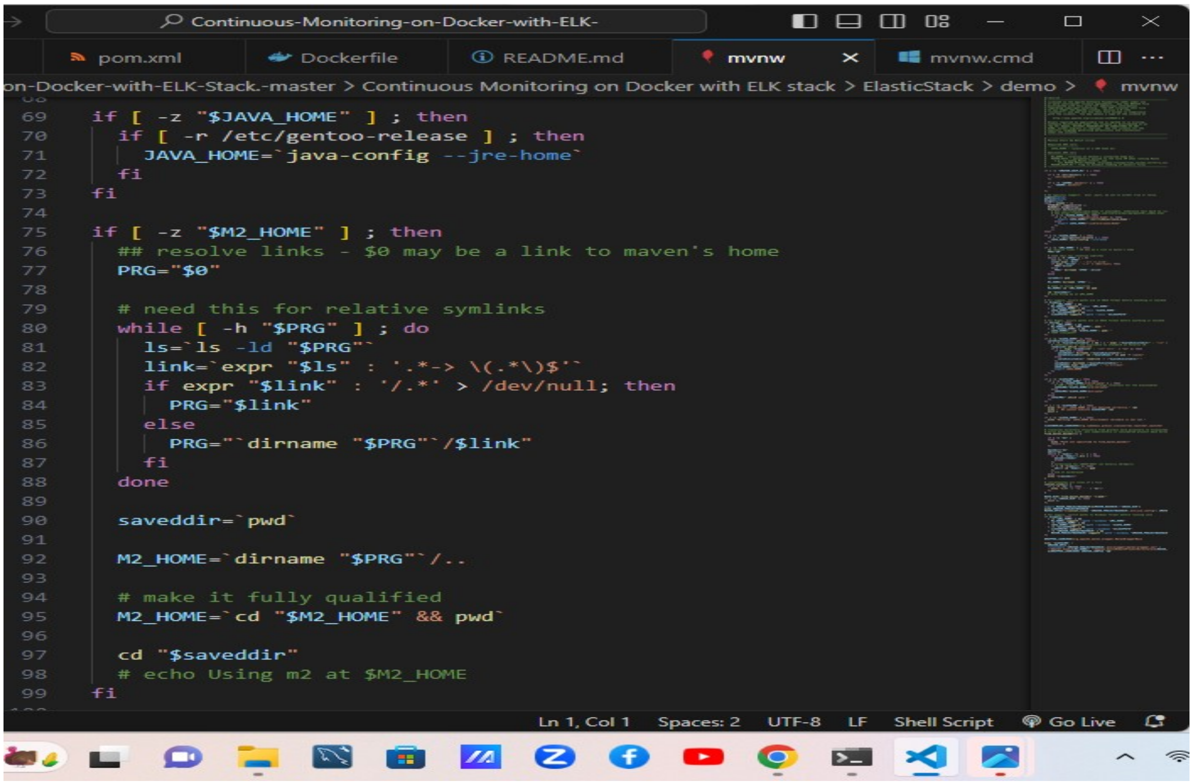
```
@if "%MAVEN_BATCH_ECHO%" == "on"  echo %MAVEN_BATCH_ECHO%

@REM set %HOME% to equivalent of $HOME
if "%HOME%" == "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")

@REM Execute a user defined script before this one
if not "%MAVEN_SKIP_RC%" == "" goto skipRcPre
@REM check for pre script, once with legacy .bat ending and once with .c
if exist "%HOME%\mavenrc_pre.bat" call "%HOME%\mavenrc_pre.bat"
if exist "%HOME%\mavenrc_pre.cmd" call "%HOME%\mavenrc_pre.cmd"
:skipRcPre

@setlocal

set ERROR_CODE=0

@REM To isolate internal variables from possible post scripts, we use an
@setlocal

@REM ==== START VALIDATION ====
if not "%JAVA_HOME%" == "" goto OkJHome

echo.
echo Error: JAVA_HOME not found in your environment. >&2
echo Please set the JAVA_HOME variable in your environment to match the
echo location of your Java installation. >&2
echo.
goto error

:OkJHome
if exist "%JAVA_HOME%\bin\java.exe" goto init
```

Mvnw screenshot



pom.xml
Dockerfile
README.md
mvnw
mvnw.cmd
ıg-on-Docker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > mvnw
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Shell Script
Go Live

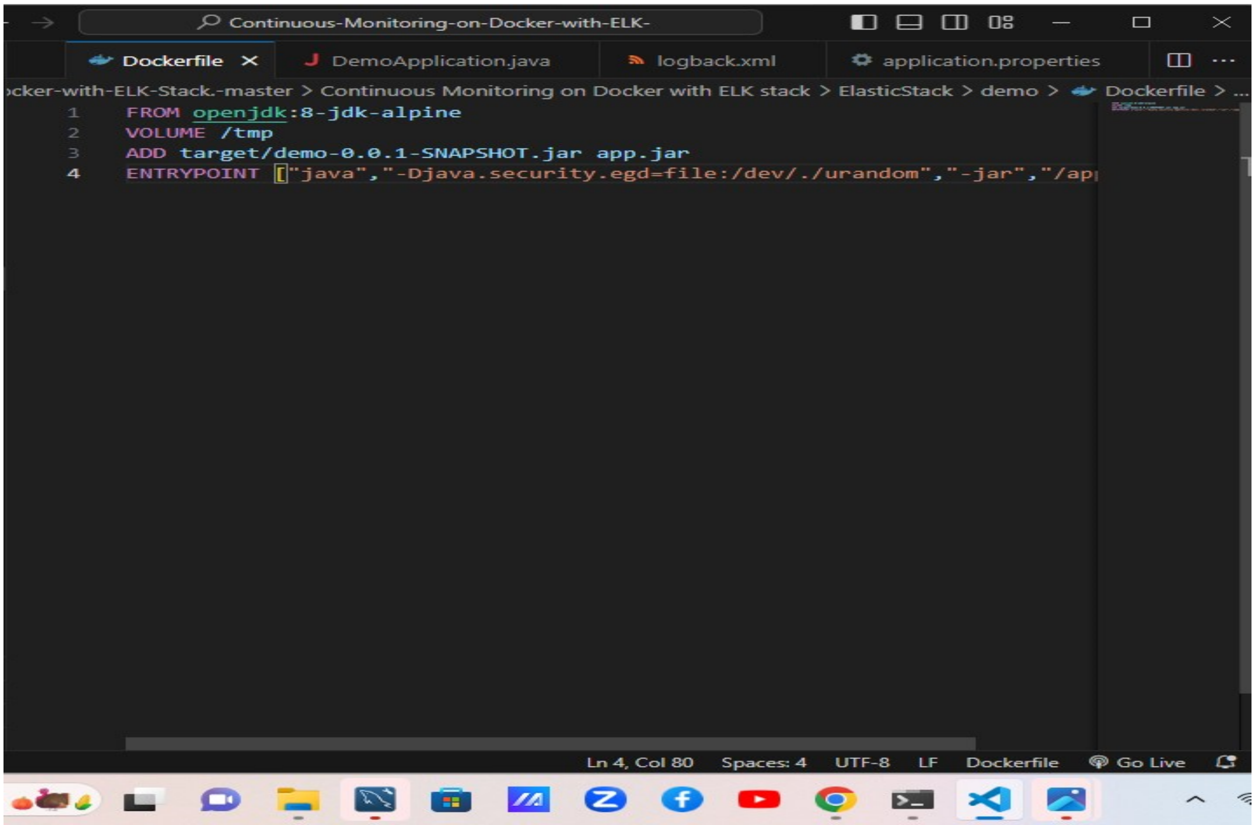
```
# For Migwn, ensure paths are in UNIX format before anything is touched
if $mingw ; then
  [ -n "$M2_HOME" ] &&
    M2_HOME="`(cd "$M2_HOME"; pwd)`"
  [ -n "$JAVA_HOME" ] &&
    JAVA_HOME="`(cd "$JAVA_HOME"; pwd)`"
  # TODO classpath?
fi

if [ -z "$JAVA_HOME" ]; then
  javaExecutable="`which javac`"
  if [ -n "$javaExecutable" ] && ! [ "`expr \"$javaExecutable\" : '\([^
    # readlink(1) is not available as standard on Solaris 10.
    readLink=`which readlink`
    if [ ! `expr "$readLink" : '\([^ ]*\)'` = "no" ]; then
      if $darwin ; then
        javaHome="`dirname \"$javaExecutable\"`"
        javaExecutable="`cd \"$javaHome\" && pwd -P`/javac"
      else
        javaExecutable="`readlink -f \"$javaExecutable\"`"
      fi
      javaHome="`dirname \"$javaExecutable\"`"
      javaHome=`expr "$javaHome" : '\(.*\)/bin'`
      JAVA_HOME="$javaHome"
      export JAVA_HOME
    fi
  fi
fi

if [ -z "$JAVACMD" ] ; then
  if [ -n "$JAVA_HOME"  ] ; then
```



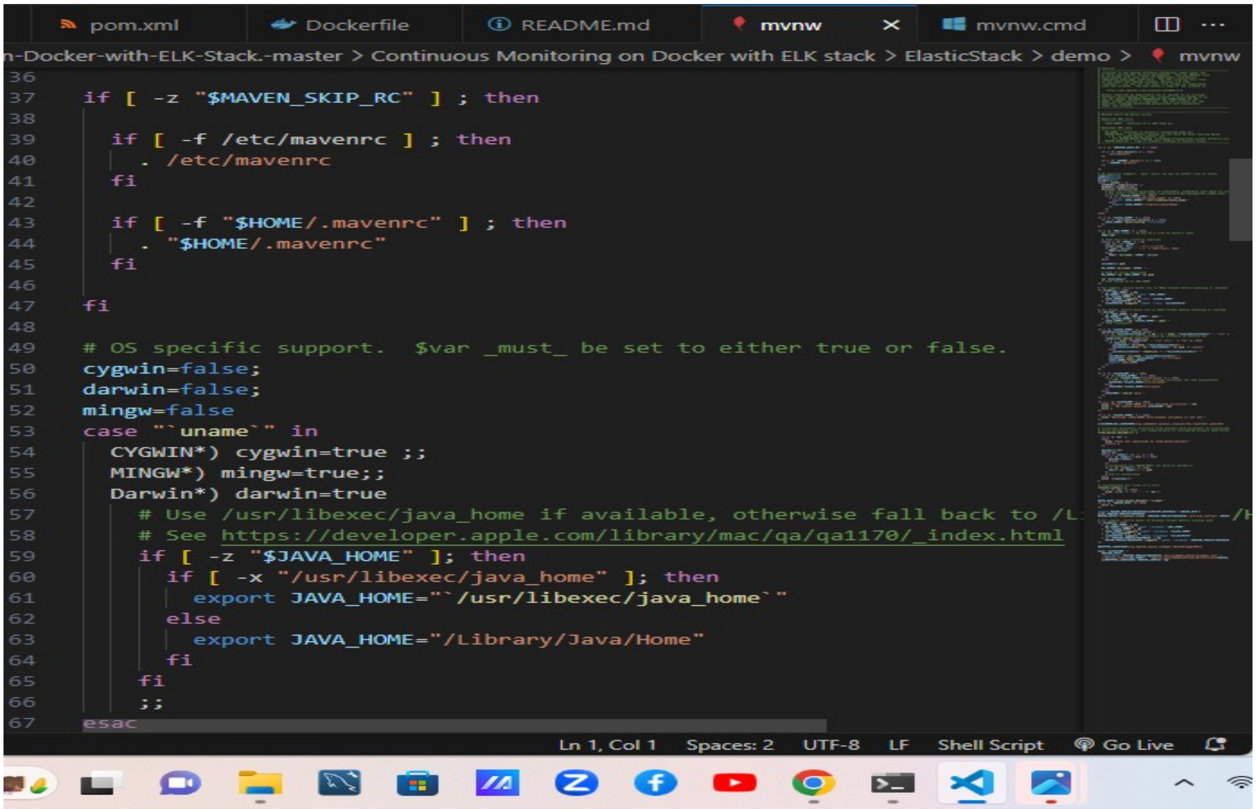
Continuous-Monitoring-on-Docker-with-ELK-
pom.xml
Dockerfile
README.md
mvnw
mvnw.cmd
on-Docker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > mvnw
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Shell Script
Go Live

```
if [ -z "$JAVA_HOME" ] ; then
  if [ -r /etc/gentoo-release ] ; then
    JAVA_HOME=`java-config --jre-home`
  fi
fi

if [ -z "$M2_HOME" ] ; then
  ## resolve links - $0 may be a link to maven's home
  PRG="$0"

  # need this for relative symlinks
  while [ -h "$PRG" ] ; do
    ls=`ls -ld "$PRG"`
    link=`expr "$ls" : '.*-> \(.*\)$'`
    if expr "$link" : '/.*' > /dev/null; then
      PRG="$link"
    else
      PRG="`dirname "$PRG"`/$link"
    fi
  done

  saveddir=`pwd`

  M2_HOME=`dirname "$PRG"`/..

  # make it fully qualified
  M2_HOME=`cd "$M2_HOME" && pwd`

  cd "$saveddir"
  # echo Using m2 at $M2_HOME
fi
```

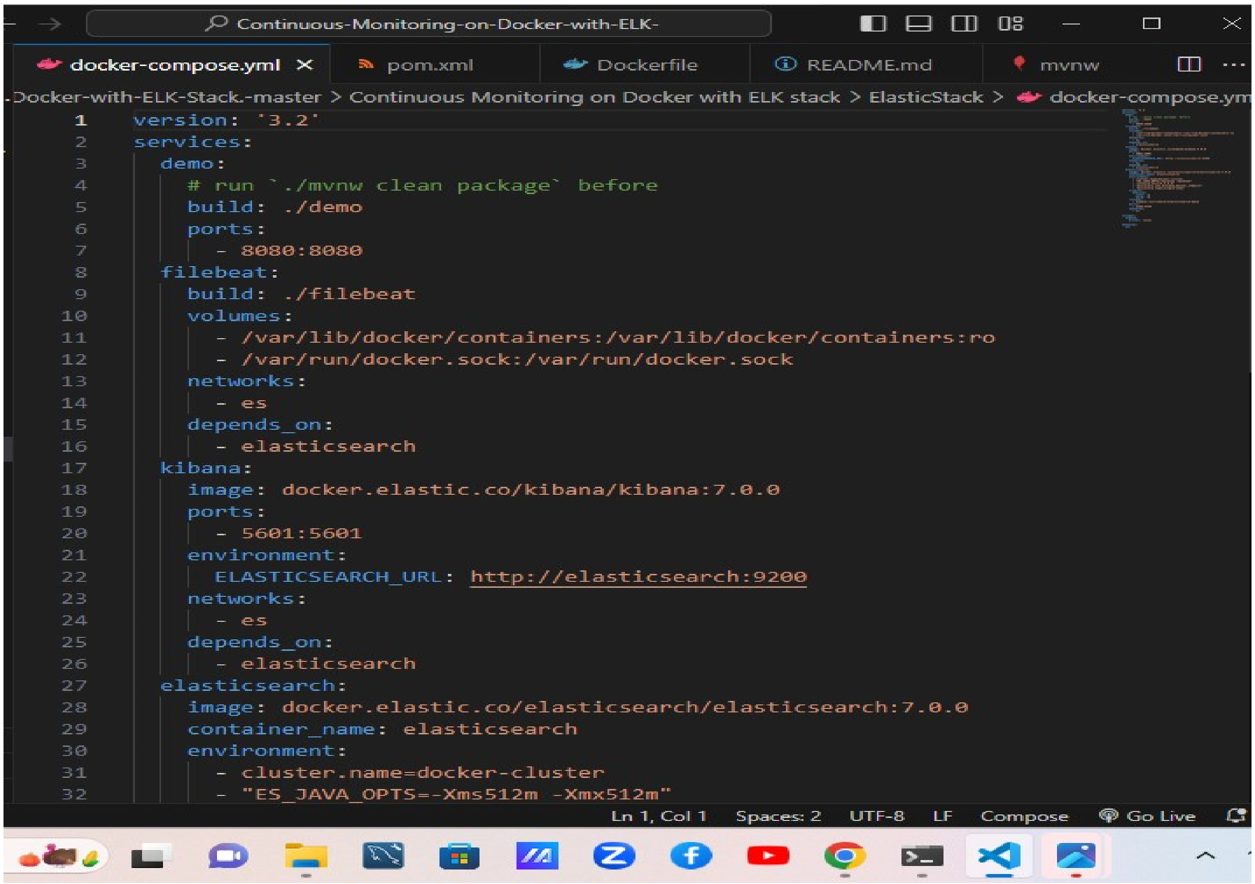
Dockerfile


Continuous-Monitoring-on-Docker-with-ELK-
Dockerfile
DemoApplication.java
logback.xml
application.properties
›cker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > Dockerfile > ...
Ln 4, Col 80
Spaces: 4
UTF-8
LF
Dockerfile
Go Live

```
FROM openjdk:8-jdk-alpine
VOLUME /tmp
ADD target/demo-0.0.1-SNAPSHOT.jar app.jar
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/ap
```

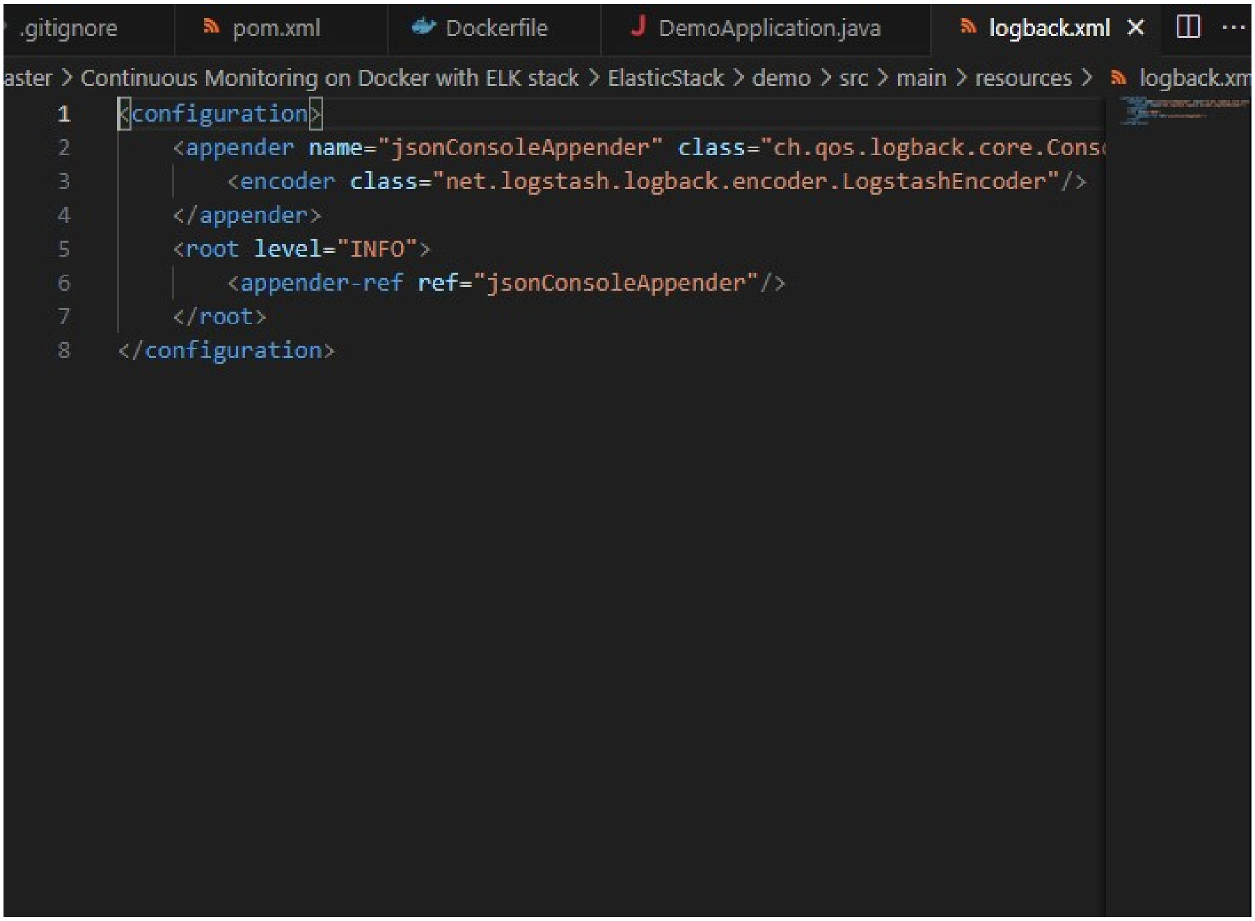

pom.xml
Dockerfile
README.md
mvnw
mvnw.cmd
n-Docker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > mvnw
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Shell Script
Go Live

```

if [ -z "$MAVEN_SKIP_RC" ] ; then

  if [ -f /etc/mavenrc ] ; then
    . /etc/mavenrc
  fi

  if [ -f "$HOME/.mavenrc" ] ; then
    . "$HOME/.mavenrc"
  fi

fi

# OS specific support.  $var _must_ be set to either true or false.
cygwin=false;
darwin=false;
mingw=false
case "`uname`" in
  CYGWIN*) cygwin=true ;;
  MINGW*) mingw=true;;
  Darwin*) darwin=true
    # Use /usr/libexec/java_home if available, otherwise fall back to /L
    # See https://developer.apple.com/library/mac/qa/qa1170/_index.html
    if [ -z "$JAVA_HOME" ]; then
      if [ -x "/usr/libexec/java_home" ]; then
        export JAVA_HOME="`/usr/libexec/java_home`"
      else
        export JAVA_HOME="/Library/Java/Home"
      fi
    fi
    ;;
esac
```


Continuous-Monitoring-on-Docker-with-ELK-
docker-compose.yml
pom.xml
Dockerfile
README.md
mvnw
Docker-with-ELK-Stack.-master > Continuous Monitoring on Docker with ELK stack > ElasticStack > docker-compose.ym
Ln 1, Col 1
Spaces: 2
UTF-8
LF
Compose
Go Live

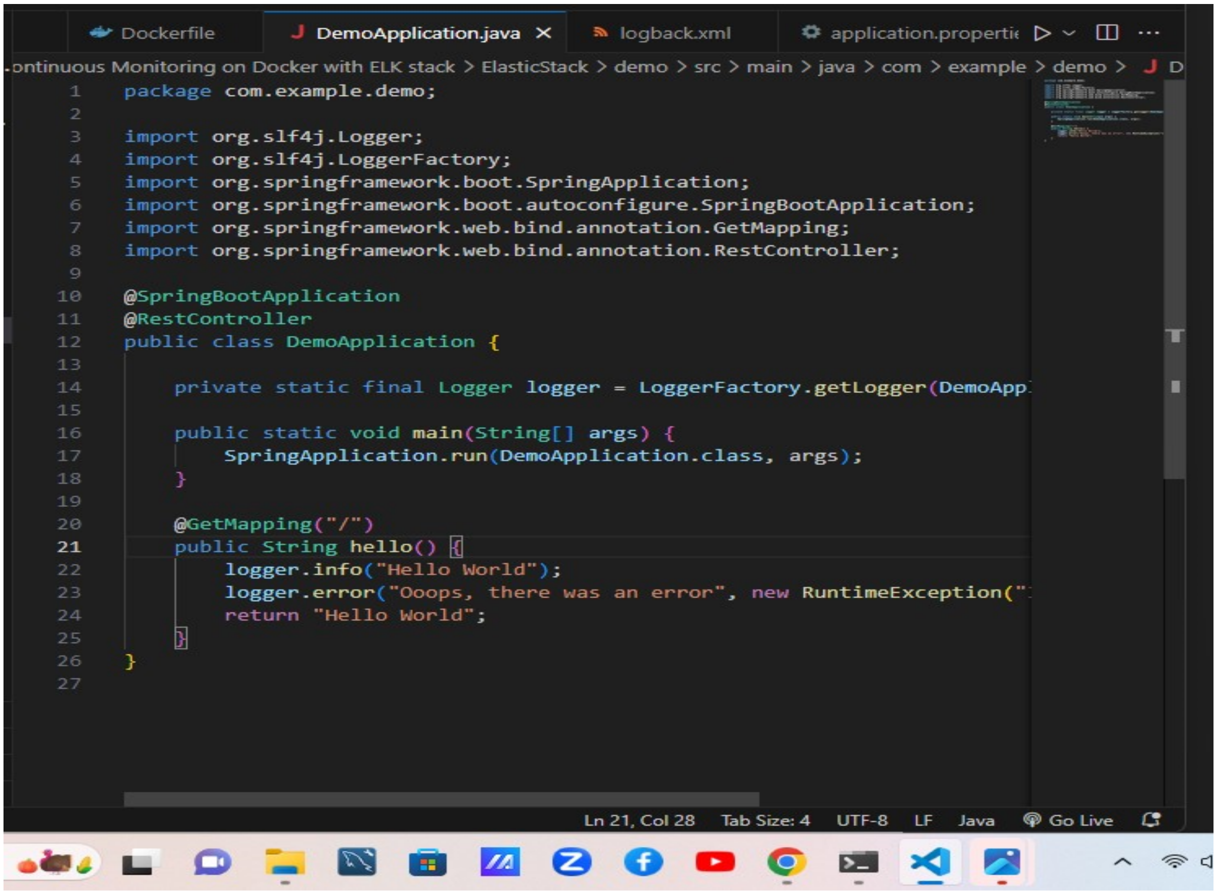
```
version: '3.2'
services:
  demo:
    # run `./mvnw clean package` before
    build: ./demo
    ports:
      - 8080:8080
  filebeat:
    build: ./filebeat
    volumes:
      - /var/lib/docker/containers:/var/lib/docker/containers:ro
      - /var/run/docker.sock:/var/run/docker.sock
    networks:
      - es
    depends_on:
      - elasticsearch
  kibana:
    image: docker.elastic.co/kibana/kibana:7.0.0
    ports:
      - 5601:5601
    environment:
      ELASTICSEARCH_URL: http://elasticsearch:9200
    networks:
      - es
    depends_on:
      - elasticsearch
  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.0.0
    container_name: elasticsearch
    environment:
      - cluster.name=docker-cluster
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
```

Logback


.gitignore
pom.xml
Dockerfile
DemoApplication.java
logback.xml
aster > Continuous Monitoring on Docker with ELK stack > ElasticStack > demo > src > main > resources > logback.xm

```xml
<configuration>
    <appender name="jsonConsoleAppender" class="ch.qos.logback.core.Cons
        <encoder class="net.logstash.logback.encoder.LogstashEncoder"/>
    </appender>
    <root level="INFO">
        <appender-ref ref="jsonConsoleAppender"/>
    </root>
</configuration>
```

Demo Application.java


Dockerfile
DemoApplication.java
logback.xml
application.propertie
ontinuous Monitoring on Docker with ELK stack > ElasticStack > demo > src > main > java > com > example > demo > D
Ln 21, Col 28
Tab Size: 4
UTF-8
LF
Java
Go Live

```java
package com.example.demo;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
public class DemoApplication {

    private static final Logger logger = LoggerFactory.getLogger(DemoApp

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

    @GetMapping("/")
    public String hello() {
        logger.info("Hello World");
        logger.error("Ooops, there was an error", new RuntimeException("
        return "Hello World";
    }
}

```